

<b>Started on</b>	Thursday, 25 April 2024, 8:42 PM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 25 April 2024, 10:48 PM
<b>Time taken</b>	2 hours 6 mins
<b>Grade</b>	<b>400.00</b> out of 400.00 ( <b>100%</b> )

Question 1

Correct

Mark 80.00 out of 80.00

Time limit	1 s
Memory limit	64 MB

### Array Manipulation

Buatlah sebuah kelas dengan nama **ArrayManipulation**. Kelas tersebut memiliki beberapa static generic method yang dapat digunakan untuk melakukan manipulasi array seperti menghasilkan array gabungan yang tersortir, mengembalikan nilai maksimal dari array, dan lain-lain.

Method-method yang dimiliki oleh kelas tersebut adalah

1. **mergeAndSortArrays**, digunakan untuk menggabungkan dan mengurutkan dua array berisikan elemen yang **"comparable"**.
2. **findMax**, digunakan untuk menemukan elemen maksimum dalam array berisikan elemen yang **"comparable"**.
3. **mapArray**, digunakan untuk menerapkan fungsi "mapper" ke setiap elemen dalam array dan mengembalikan array hasilnya. Array berisikan elemen yang **"comparable"**.
4. **filterArray**, digunakan untuk mengembalikan array baru yang berisi elemen dari array masukan yang memenuhi kondisi yang ditentukan oleh suatu fungsi **"predicate"**.

Disediakan kerangka kelas [ArrayManipulation.java](#), lengkapi implementasi method-methodnya. Di dalam kerangka tersebut juga disediakan main function untuk membantu debugging. Kumpulkan [ArrayManipulation.java](#) yang sudah dilengkapi methodnya

Java 8

 [ArrayManipulation.java](#)

Score: 80

Blackbox

Score: 80

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.07 sec, 28.55 MB
2	10	Accepted	0.07 sec, 28.43 MB
3	10	Accepted	0.06 sec, 28.33 MB
4	10	Accepted	0.07 sec, 28.49 MB
5	10	Accepted	0.07 sec, 28.81 MB
6	10	Accepted	0.07 sec, 28.42 MB
7	10	Accepted	0.06 sec, 29.00 MB
8	10	Accepted	0.07 sec, 28.79 MB

## Question 2

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Lengkapi [GameKata.java](#). Gunakan Java API Map untuk menyelesaikan persoalan ini.

Tito dan Wiwid sedang bermain sebuah game kata. Tito dan Wiwid menuliskan masing-masing N dan M jumlah kata pada kertas milik mereka. Pada akhir permainan, mereka akan mencocokkan kata yang mereka tulis di kertas.

Program akan menuliskan kata yang sama dari kedua orang, dipisahkan oleh garis baru. Note: Kata yang dicocokkan **case-sensitive**. Urutan dari kata yang dituliskan oleh program tidak perlu urut.

Contoh input-output sebagai berikut:

Contoh Input	Contoh Output
// Tito Viel Adalah Orang Imba // Wiwid Aku Adalah Orang Imba	Adalah Orang Imba
// Tito Budi Budi Adalah Budi // Wiwid Budi Bermain Bola Adalah	Budi Adalah
// Tito Saya Mengantuk // Wiwid Ingin Tidur	

Lengkapi dan kumpulkan file **GameKata.java**.

Java 8

 [GameKata.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.12 sec, 28.94 MB
2	10	Accepted	0.09 sec, 28.84 MB
3	10	Accepted	0.08 sec, 28.73 MB

No	Score	Verdict	Description
4	10	Accepted	0.07 sec, 29.73 MB
5	10	Accepted	0.06 sec, 28.45 MB
6	10	Accepted	0.07 sec, 28.44 MB
7	10	Accepted	0.07 sec, 27.84 MB
8	10	Accepted	0.07 sec, 28.42 MB
9	10	Accepted	0.07 sec, 28.71 MB
10	10	Accepted	0.09 sec, 28.73 MB

Time limit	1 s
Memory limit	64 MB

Seorang Komandan ingin melakukan simulasi untuk membariskan anggotanya untuk kepentingan apel istimewa dalam rangka peringatan hari jadi negara. Lantas komandan melakukan simulasi barisan tersebut dengan memanfaatkan ilmu programming Java yang ia pelajari dari seorang hacker yang ia pernah tangkap 4 tahun silam. Komandan melakukan percobaan barisan dengan memanfaatkan API Java, yaitu `ArrayList` dan `LinkedList`. Kedua kelas ini memiliki implementasi yang berbeda. Tolong bantu komandan untuk melakukan simulasi terhadap anggotanya.

Buatlah sebuah kelas **Pasukan**:

- memiliki private attribute list dengan tipe `List<Integer>`
- memiliki private attribute `n` dengan tipe `int`
- memiliki public method `void reset()`, yang memanggil `clear()` dari list dan mengisi list dengan `n` buah bilangan 0.
- memiliki public method `long get(int idx)` yang jika dipanggil:
  - Mereset list dengan memanggil method `reset`
  - Mencatat waktu sekarang (start) dalam milisekon
  - Mengambil indeks ke `idx` dari list
  - Mencatat waktu sekarang (finish) dalam milisekon
  - Mengembalikan waktu `finish - start`
- memiliki public method `long del(int idx, int t)` yang jika dipanggil:
  - Mereset list dengan memanggil method `reset`
  - Mencatat waktu sekarang (start) dalam milisekon
  - Menghapus elemen ke-`idx` dari list sebanyak `t` kali
  - Mencatat waktu sekarang (finish) dalam milisekon
  - Mengembalikan waktu `finish - start`

Lihat contoh [Main.java](#). Untuk mengukur waktu, Anda wajib menggunakan [Util.java](#)

Kumpulkan **Pasukan.java**

Java 8

 [Pasukan.java](#)

Score: 120

Blackbox

Score: 120

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.07 sec, 28.92 MB
2	20	Accepted	0.07 sec, 28.70 MB
3	20	Accepted	0.07 sec, 28.45 MB
4	20	Accepted	0.07 sec, 27.91 MB
5	20	Accepted	0.07 sec, 28.19 MB
6	20	Accepted	0.07 sec, 28.04 MB

## Question 4

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Diceritakan Putri dari Kerajaan Permen mengoleksi sejumlah hewan eksotis seperti singa dan panda. Sebagai bentuk kemurahan hatinya, sang Putri ingin membuat sebuah kebun binatang supaya penduduk Kerajaan Permen dapat melihat koleksi hewan yang dimilikinya. Wujudkanlah keinginan sang Putri dengan membuat kelas Enclosure, yaitu kandang yang dapat menyimpan hewan-hewan milik sang Putri.

Diberikan [Animal.java](#), [Lion.java](#), dan [Panda.java](#)

Lengkapilah [Enclosure.java](#)

Contoh output untuk method describe:

Kode program	Output
<pre>Enclosure enc1 = new Enclosure(); Panda panda1 = new Panda("po", 5); Panda panda2 = new Panda("po2", 5); enc1.addAnimal(panda1); enc1.addAnimal(panda2); enc1.describe();</pre>	Kandang berisi 2 ekor hewan: 1. po, spesies Pandamus Maximus, umur 5 tahun 2. po2, spesies Pandamus Maximus, umur 5 tahun
<pre>Enclosure enc1 = new Enclosure(); enc1.describe();</pre>	Kandang ini kosong...

Java 8

 [Enclosure.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	8	Accepted	0.34 sec, 27.86 MB
2	8	Accepted	0.32 sec, 28.96 MB
3	8	Accepted	0.14 sec, 28.77 MB
4	8	Accepted	0.10 sec, 28.16 MB
5	8	Accepted	0.12 sec, 27.86 MB
6	8	Accepted	0.15 sec, 28.77 MB
7	8	Accepted	0.10 sec, 29.04 MB
8	8	Accepted	0.11 sec, 29.82 MB
9	8	Accepted	0.07 sec, 26.47 MB
10	8	Accepted	0.07 sec, 29.95 MB

No	Score	Verdict	Description
11	8	Accepted	0.07 sec, 28.16 MB
12	12	Accepted	0.07 sec, 28.65 MB

[← Latihan Praktikum 4](#)

Jump to...



[Slide Tutorial 5 ►](#)